Preprint
UCRL-JC-147384

Multi-Resolution Modeling of Large Scale Scientific Simulation Data

C.H. Baldwin, G.M. Abdulla, C.J. Critchlow

This article was submitted to The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada, July 23-26, 2002

February 26, 2002

U.S. Department of Energy

Lawrence
Livermore
National
Laboratory

This work was performed under the auspices of the United States Department of Energy by the University of California, Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

Multi-Resolution Modeling of Large Scale Scientific Simulation Data

Chuck Baldwin
baldwin5@llnl.gov

Ghaleb Abdulla
abdulla1@llnl.gov

Terence Critchlow
critchlow1@llnl.gov

Center for Applied Scientific Computing
Lawrence Livermore National Laboratory
Livermore, CA 94551

ABSTRACT

Data produced by large scale scientific simulations, experiments, and observations can easily reach tera-bytes in size. The ability to examine data-sets of this magnitude, even in moderate detail, is problematic at best. Generally this scientific data consists of multivariate field quantities with complex inter-variable correlations and spatial-temporal structure. To provide scientists and engineers with the ability to explore and analyze such data sets we are using a twofold approach. First, we model the data with the objective of creating a compressed yet manageable representation. Second, with that compressed representation, we provide the user with the ability to query the resulting approximation to obtain approximate yet sufficient answers; a process called ad-hoc querying. This paper is concerned with a wavelet modeling technique that seeks to capture the important physical characteristics of the target scientific data. Our approach is driven by the compression, which is necessary for viable throughput, along with the end user requirements from the discovery process. Our work contrasts existing research which applies wavelets to range querying, change detection, and clustering problems by working directly with a decomposition of the data. The difference in this procedures is due primarily to the nature of the data and the requirements of the scientists and engineers. Our approach directly uses the wavelet coefficients of the data to compress as well as query. We will provide some background on the problem, describe how the wavelet decomposition is used to facilitate data compression and how queries are posed on the resulting compressed model. Results of this process will be shown for several problems of interest and we will end with some observations and conclusions about this research.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Scientific databases*; G.1.2 [**Numerical Analysis**]: Approximation—*Wavelets and fractals*; E.4 [**Data**]: Coding and Information Theory—*Data compaction and compression*

General Terms

Algorithms, Management, Performance

Keywords

scientific data processing, data modeling, wavelets, compression

1. INTRODUCTION

For many years knowledge discovery and data mining techniques have been applied to data obtained from the fields of information science, business, and the social sciences. Consumer modeling, financial market trends, internet traffic modeling, web searching, and census data analysis are a few examples of these broad disciplines where one can find work in the application of knowledge discovery and data mining techniques. The use of machine learning, pattern recognition, and statistical modeling on data obtained from experiments, observations and simulations is becoming of great interest to scientists and engineers. It has been noted [14] that the flood of data inherent in large scale scientific simulations or vast observational catalogues has led scientists and engineers to explore better, more efficient, ways of understanding the data being produced. Many techniques in knowledge discovery and data mining are currently being explored by researchers to help address this problem [7]. This paper is devoted to one aspect of this important growth area of knowledge discovery and data mining. Our problem is one of effectively compressing large scale scientific simulation data (measured in tera-bytes) and allowing a scientist to query the compressed data in a fraction of the time a similar query operation would take on the original data. In order to achieve this throughput objective, we are exploring data modeling techniques that significantly reduce the overall size of the data while effectively maintaining much of the important physical characteristics of the data – thereby defining an ad-hoc query process. We are exploring many techniques from mathematical and statistical modeling to effectively capture the important yet relevant behavior of the data. The choice of a particular data model vary depending on the characteristics of the data we wish to model, the particular properties of the modeling technique, and the types of queries we wish to resolve.

Multi-resolution techniques are based on the idea that data exhibits effects based on temporal or spatial scale and seeks to efficiently model that behavior through simple, and efficient filtering operations [9, 19]. Wavelets apply two sets of filters to produce reduced size *scaled* and *detail* versions of the original data; the filters are again repeatedly applied "downwards" to the scaled versions to produce a final smooth scale approximate of the data and a hierarchy of detail data of increasing size. The whole process is reversible, in that equivalent filters can be applied and combined at each level "upwards" to recover the original data. Our reduced set models are derived from the detail coefficients. Dropping coefficients, not storing them, has a measurable effect on the reconstructed approximate to the original data. We are experimenting with procedures that effectively drop coefficients, thereby reducing the model size, while maintaining structure that is appropriate for ad-hoc querying of the result. From the compressed data, we are also allowing further user defined reconstructions of the original data – this constitutes the query aspects of the system for wavelets and makes the approach unique. We define a set of queries based on abstract qualities or concrete quantities of the stored coefficients. Reconstruction based on a specified number of important levels in the stored decomposition or reconstruct with defined relative error, with respect to the original model, are examples of these types of queries. Our goal with this query paradigm is to allow a scientist to effectively use the speed and characteristics of multi-resolution techniques while allowing him or her to explore large scale data without becoming an expert in multi-resolution theory.

1.1 Related Work

Multi-resolution techniques, specifically wavelets, have been used for many years as effective modeling tools for data derived from signal and image processing applications [11, 19, 21]. Compression, denoising, and change detection [8, 13] are several examples of particular problems in signal and image analysis where multi-resolution techniques have proven to be effective. As an example of this trend, the newest image compression standard, JPEG2000, utilizes wavelet based techniques [10, 20]. Multi-resolution based paradigms have also been shown to be great promise in knowledge discovery and data mining applications for data obtained from astronomical observation, specifically clustering objects in large scale sky surveys [12]. In the recent past, multi-resolution algorithms have been introduced for particular clustering problems. The *WaveCluster* [17] approach maps the data onto a multi-dimensional grid, defining a feature space, and applies a wavelet transform to obtain clusters of spatial databases. In an effort to address data-sets of high dimensionality, such as multi-media and image databases, a wavelet based clustering algorithm *HyperWave* [23] has also been introduced. Additionally, and more related to the problem we are addressing, wavelets have been successfully applied to traditional data querying applications [15]. For fast responses to range sum queries researchers [2] have developed a wavelet based approach for approximate query processing. In this work the data is mapped to a relational table, which is compressed and used to resolve *select*, *project*, and *join* operations. A progressive technique which maps the query, along with the data, to the wavelet domain for query resolution has been introduced by Shahabi, Chung and Safar [16]. This technique is more like our work but does not a-priori compress the data set to an approximation. Wavelets have also proved useful for indexing into large time series databases [6]. The idea behind this work is to treat the n-component time series of a target database as objects in n-dimensional feature space and apply a wavelet transform to the resulting feature space. Once the transform is done only a small subset of the coefficients are used to represent each time series, reducing the dimensionality of the original data significantly. Wavelets have also been used as a basis for answering surprise and trend queries in time series [15]. In this research a wavelet transform is applied to the time series data and using the results of the wavelet decomposition are stored in a level-wise tree. The trend or surprise queries are posed and answered by reconstructing the data by using only the levels of the tree that are appropriate for the query. Our problem, though different, can benefit from the same basic tenets fundamental to these other contributions – that is the analysis of data at varying scales. In general, scientific data from experiment, observation, or simulation exhibits this "scale behavior" Specific to our application, large scale time dependent three-dimensional simulations are modeled with locally discretized difference equations and tend to produce temporal and spatial correlations in the discrete data.

1.2 Contributions

Our contributions from this work are two-fold. First in the application of wavelets to model and compress the kinds of scientific data we are targeting and second, in the techniques for querying that resulting compressed model. The data we are interested in is large scale multivariate field quantities from simulations, experiments, or observations. Typical quantities found in these simulations are fundamental or derived physical quantities such as temperature, pressure, velocity, vorticity, or entropy. Although some work has been done in multivariate wavelet transforms in the recent past the concept is an ongoing research topic in the wavelet community and one which we are exploring. In order to address this issue we have taken a particular mathematical approach to modeling and compressing multivariate data which allows us to adequately treat the effects of such data on the resulting model and, more importantly, effectively compress the data. We believe that the simple yet sound solution approach is of interest to researchers working with multivariate data. We have also established a manner in which we allow queries to occur on the resulting compressed data. We are directly querying the compressed wavelet transform data rather than the original data itself. This means that our queries are posed with reguard to the wavelet transform of a temperature field is as opposed to the temperature field itself, for instance. Focusing on the latter would necessitate either mapping equivalent queries to the domain of the wavelet transform or reverse the transform in some intelligent fashion to obtain approximations to or subsets of the field data itself. Querying the wavelet transform data itself has a practical problem associated with it, namely understanding how to think in the domain of the wavelet transform data rather than the intuitive domain of the field data. To address this issue we define specific queries (with associated semantics) that will allow a particular coupled reconstruction from the wavelet transform data. It should be noted that we are exploring other modeling techniques which lend themselves to resolving other types of queries, such as range based queries, however for this work we will concentrate on following this model/query framework.

2. ALGORITHM DESCRIPTIONS

Figure 1 illustrates the simplified diagram of our ad-hoc query system (known as AQSIM) [1]. The important points for this discussion are that a modeling technique (in this case wavelets) will create a simplified model file and a data reconstructor will use that model file and a user defined query to create an approximate representation for the original model data. The simplified model is written to disk in a pre-processing step (shown on the right side of the figure) which has few processing time constraints, the ad-hoc query phase (shown on the left side of the figure) is under user control and does have constraints based on response time. We use this fact to compute and store the compressed wavelet model in an advantageous form for the ad-hoc query phase.

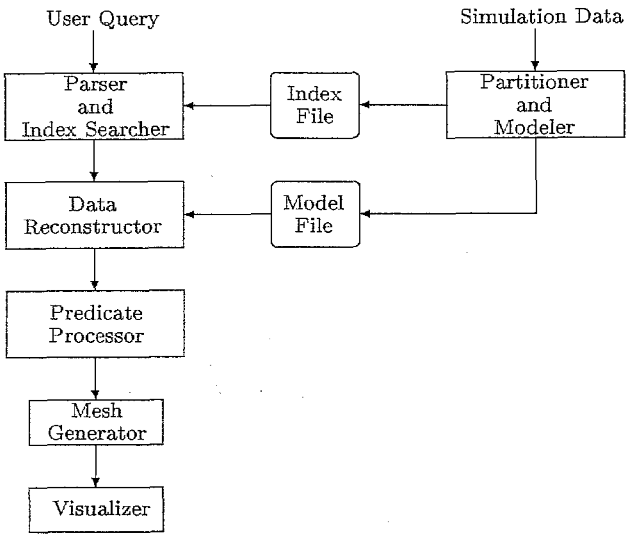

Figure 1: AQSIM system diagram

This section will address first how we create the wavelet model for our original simulation data in the pre-processing phase and second how we reconstruct the approximate simulation data in the query phase.

2.1 Basic Wavelet Theory

There are many excellent books and papers on the theory of wavelets [3, 9, 19], and the general application of multi-resolution analysis techniques to various problem domains [11, 18, 22]. We shall not reproduce that body of work here but would like to give a brief review of multi-resolution analysis with traditional orthonormal wavelets to describe how and why we are developing the algorithms in our system. We begin with a basic development of general wavelet theory, first in the continuous case then moving to the discrete. The idea of a continuous wavelet begins with a *scaling function* ϕ and a *wavelet function* ψ which satisfies the *dilation equations* :

$$\phi(t) = \sqrt{2}\sum_j c_j\phi(2t-j),$$

and

$$\psi(t) = \sqrt{2}\sum_j w_j\phi(2t-j).$$

The filter coefficients $\{c_j\}$ and $\{w_j\}$ determine the smoothness, orthogonality, vanishing moments and compactness properties of the resulting functions. These scaling and wavelet functions along with their dilates and translates defined by :

$$\phi_j^k(t) = \phi(2^k t - j)$$

and

$$\psi_j^k(t) = \psi(2^k t - j)$$

establishes a sub-space V^0 and a sequence of sub-spaces $\{W^k\}_{k=0}^{\infty}$ which together form a direct sum decomposition of L^2 (the space of square integrable functions) in the following sense :

$$L^2 = V^0 \bigoplus_{k=0}^{\infty} W^k. \tag{1}$$

This direct sum decomposition allows any square integrable function to be written exactly as a sum of the projections of that function onto V^0 and each of the W^k. There are many different choices for the filter coefficients found in the literature which form these (bi-orthogonal) basis. For our work we will be using Daubechies orthonormal wavelets [3]. The familiar Haar wavelet is a simple example of an orthonormal wavelet which is considered among this family.

We are concerned with discrete data (traditionally signals) and not continuous data in our work so the concepts introduced earlier have to be extended to work in the case of discrete data. Multi-resolution for continuous functions extends to the discrete case in a analogous fashion and follows from traditional literature in the signal processing community [11, 19]. The idea behind wavelet decompositions of signals is that given a signal f of size N a pair of reduced size (coarser) discrete signals s and d defined on a dyadic coarsing[1] of the original domain can be computed – analogous to the previous continuous discussion. The computation is done by applying a low-pass linear filter G (followed with down-sampling by a factor of 2) and a high-pass linear filter H (also followed with down-sampling by a factor of 2) to the original signal f in a process known as *analysis*. The two signals s and d represent coarse low-pass and high-pass filters of the original signal. An important property of these multi-resolution algorithms is that the original signal f can be reconstructed from the reduced size low-pass and high-pass filtered signals s and d – in a process known as *synthesis*. In this process, a low-pass filter G^* (preceded by up-sampling by a factor of 2) and a high-pass filter H^* (also preceded by up-sampling by a factor of 2) are applied to s and d to produce two signals that can be combined with simple addition to produce the original signal f. The two synthesis filters G^* and H^* are intrinsically related to the original analysis filters G and H and their construction, along with their specific attributes, is the result of many

[1]A dyadic coarsening refers to the fact that two elements of the fine domain data are combined into one element of the coarse domain data.

early papers in the field [3, 9]. The computational complexity of the above filtering and sampling operations is $O(N)$ since the number of coefficients in both filters a constant ($\ll N$ by design). This decomposition/reconstruction property is known as "perfect reconstruction" and is shown in figure 2.

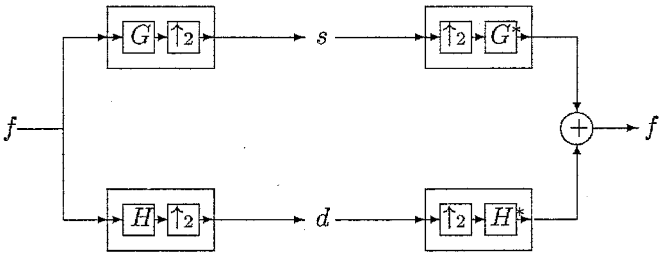

Figure 2: Perfect Reconstruction Property

In terms of finite filters, the perfect reconstruction property amounts to the the following mathematical identity :

$$G^*G + H^*H = I,$$

where G, H, G^*, and H^* are the analysis and synthesis filters from above and I is the identity filter. The idea behind a multi-resolution analysis of a signal is to use the same decomposition operation on the filtered and down-sampled scaled signal s at subsequent levels in a recursive fashion. It is easy to see that if the perfect reconstruction property holds at a single level then the whole process will hold for a hierarchy of levels. The output of this process produces a small set of J smooth scaling coefficients at a very coarse level, say L,

$$\{s_j^L\}_{j=1}^J$$

and the details from all levels, $1 \leq l \leq L$,

$$\{\{d_j^l\}_{j=1}^{J_l}\}_{l=1}^L.$$

where J_l represents the indices of coefficients on level l. It can be shown (and follows intuitively from equation 1) that the original signal f has a representation in this discrete wavelet basis as :

$$f(t) = \sum_{j=1}^{J} s_j^L \phi_j^L(t) + \sum_{l=1}^{L} \sum_{j=1}^{J_l} d_j^l \psi_j^l(t), \tag{2}$$

where $\phi_j^L(t) = \phi(2^L t - j)$ is the smooth scaling function at the coarsest level and $\psi_j^l(t) = \phi(2^l t - j)$ are the detail scaling functions at the intermediate levels. Equation 2 represents the full wavelet model for our process with respect to a particular wavelet filter. For efficiency reasons the reconstruction of the original function f is done by the up-sampling and filtering operations described earlier. The representation in equation 2 is useful for some direct computations as well as observing that the size of the wavelet coefficients gives an exact computation of the size of the function f :

$$\|f\|^2 = \sum_{j=1}^{J} \|s_j^L \phi_j^L\|^2 + \sum_{l=1}^{L} \sum_{j=1}^{J_l} \|d_j^l \psi_j^l\|^2,$$

which equals from the definition of ϕ_j^L and ψ_j^l,

$$\|f\|^2 = \sum_{j=1}^{J} \|\sqrt{2}^L s_j^L\|^2 + \sum_{l=1}^{L} \sum_{j=1}^{J_l} \|\sqrt{2}^l d_j^l\|^2. \tag{3}$$

2.2 Data Compression

We use the results of equation 3 as a guide in compressing and organizing the wavelet model of the data. In fact, this gives an intuitive yet very effective method of compressing data (measured with an l^2 norm) – namely keep the coefficients with largest absolute value, weighted by a factor involving their level. A more mathematically rigorous development of this idea has been done in the literature [4, 5, 21]. This insight into the relation to the coefficients and their individual contribution to the global error actually gives three methods to store compressed model files, two of which begin with sorting the level weighted coefficients (largest to smallest) and another that doesn't require sorting :

- Choose sorted coefficients until a user specified *total number of coefficients* is achieved, thereby assuring that a prescribed model file size is achieved.
- Choose sorted coefficients until a user specified *relative error* is achieved, thereby assuring that a prescribed model relative error is achieved.
- Choose unsorted coefficients that are larger than a user specified *coefficient size.*

Figure 3: Methods for Compressing a Wavelet Transform

We are researching all three strategies but have so far opted to use the first in order to effectively address model file size. It should be noted that the sorting procedure used in the first two methods above has complexity $O(N \lg(N))$ in the number of coefficients N. This is an acceptable cost for construction of the wavelet model in our pre-processing stage. Due to sub-additivity we can rewrite the formula given by equation 2 as :

$$f(t) = \sum_{j \in \mathcal{J}_\Lambda} s_j \phi_j^L(t) + \sum_{(\ell,j) \in \mathcal{J}} d_j^\ell \psi_j^\ell(t), \tag{4}$$

where the set $\mathcal{J}_\Lambda$ represents all the indices at the coarsest level and the set $\mathcal{J}$ represents the (ℓ, j) tuples composing the indices on all other (non-coarse) levels. Then the three selection methods amount to selecting a subset, say $\mathcal{J}^*$, of $\mathcal{J}$ that satisfies one of the sorting or non-sorting criteria above. The l^2 error is easily computable as the weighted size of the coefficients left out of the selection set. The above development was in terms of single variable functions and the sorting key can naturally be chosen to be the weighted coefficient size. For multivariate functions, with which we are concerned, there is a dearth of research on the general subject of multivariate or vector multi-resolution analysis. Our solution approach is to incorporate the multivariate analysis solely into the sorting and selection procedure rather than research and develop new multivariate wavelet transforms. To do this we first perform a standard single variable transform on each variable of the data as described above. Then if we label the individual transform coefficients with their multivariate component $c = 1 \dots m$ as :

$$\{\{d^l_{c;j}\}_{j=1}^{J_l}\}_{l=1}^{L}$$

and form a equivalent to the real multivariate transform coefficient as :

$$\vec{d^l_j} = (d^l_{1;j}, \dots, d^l_{m;j})$$

for each level $l = 1, \dots, L$ and level index $j = 1, \dots, J_l$. We then use a weighted multivariate l^2 norm :

$$||(v_1, \dots, v_m)||^2 = \sum_{i=1}^{m} \omega_i |v_i^2|$$

to find a size (or importance) estimate for the coefficients and use that as a sort key. The weights ω_i are positive and have the property that :

$$\sum_{i=1}^{m} \omega_i = 1.$$

The simple weights $\omega_i = 1/m$, giving equal weight to each component, is the most natural choice and currently the ones we use. However it is not illogical or difficult to use some statistical measures of the coefficients themselves to derive a more appropriate non-linear weighting scheme. Once the coefficients are chosen the resulting coefficients along with their significance ordering obtained from the sorting are saved to disk. This compressed model file represents a starting point upon which ad-hoc queries are performed.

2.3 Queries on the Compressed Data

The reconstruction of an approximate representation of the original data in the query resolution phase (the left side of figure 1) is performed under more interactive time constraints. As mentioned we store the wavelet coefficients in the model file with their sorting order and utilize this information, currently, to provide some additional query processing for the end user. This information can also be incorporated into a progressive reconstruction which is controlled by the user and provides a more visual metric to conclude when a reconstructed approximate is "good enough". The queries that we provide are ultimately queries about the quality, quantity, or possibly spatial location of the stored wavelet coefficients themselves. This approach makes the data compression a discovery process, where the compression hopefully removes unwanted noise or homogenizes redundant information so that discovery of useful facts can be achieved. This connection between compression and knowledge discovery has been noted by Ramakrishnan and Grama [14].

The collection of wavelet queries that we are currently working on are shown in figure 4. The figure describes in words the semantics of the queries we are interested in.

The first query in figure 4 will use the complete model of the data to build the best approximate to the original data. This, in effect, just uncompresses the data for the user. The second query in figure 4 uses the pre-sorted coefficients to reconstruct an approximate to the original data with a user specified percentage of the available data. The third query in figure 4 uses the pre-sorted coefficients to reconstruct an approximate to the original data with a specified relative error (as measure against the original data). The second and third queries can be implemented in a progressive fashion; namely, the coarse scale smooth data can be displayed to the user and as the sorted coefficients are added back to the approximate data (using the representation formula in equation 2) the display of the data can be updated to reflect this gained accuracy. This process can also be interruptible. The progressive display or interruptibility is another of our long range research goals for the ad-hoc query system. The fourth query in figure 4 uses the coefficients from the most important levels to reconstruct an approximate to the original data. The notion of important levels is inherent in the representation formula in equation 3. By computing the combined total of the weighted coefficients on the different levels of the compressed model a relative merit for adding each levels coefficients can be compared and used. The fifth query in figure 4 represents a point wise reconstruction, again using the representation formula. By establishing containment of a given spatial location on each of the scaling and wavelet functions a pointwise reconstruction of the original data can be performed by using equation 2.

1. Reconstruct using all the coefficients from the stored wavelet decomposition.
2. Reconstruct by further choosing the most significant wavelet coefficients based on a user supplied percentage.
3. Reconstruct by choosing the wavelet coefficients that produce an approximate with a user supplied relative error.
4. Reconstruct using only the most significant levels of the wavelet decomposition in the model file.
5. Reconstruct using coefficients that effect a given spatial location.

Figure 4: Queries Relevant to the Wavelet Model

3. EXAMPLES

To illustrate the basic ideas of this research we present two examples of how the process works. Our first example is a simple univariate time series. Recall that our target data is multivariate but the same ideas and algorithms hold in the univariate case. Figure 5 shows the values of the *Standard & Poors 500* for the year 2001 with respect to the stock market trading day (a value we are treating as a uniform set of integers). We first perform a wavelet transform using a simple Haar orthonormal wavelet. We next create the compressed data file, a 50 percent compression ratio is established by choosing to store only half of the wavelet coefficients. This results in a compressed approximation with 0.337% global relative (l^2 error), figure 6 shows what that compressed time series looks like by simply uncompressing it. The three additional figures show what additional reconstruction queries on the compressed approximation results in. Figure 7 is using about 33% of the original coefficients (about 66% of the compressed coefficients) and the resulting reconstruction has a global relative error of .590%. Figure 8 is using about 25% of the original coefficients (about 50%

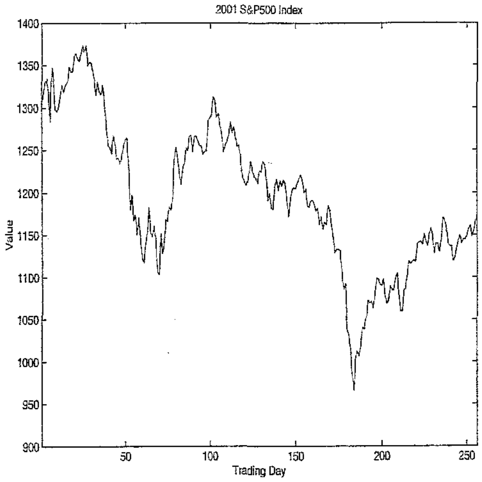

Figure 5: Original dataset.

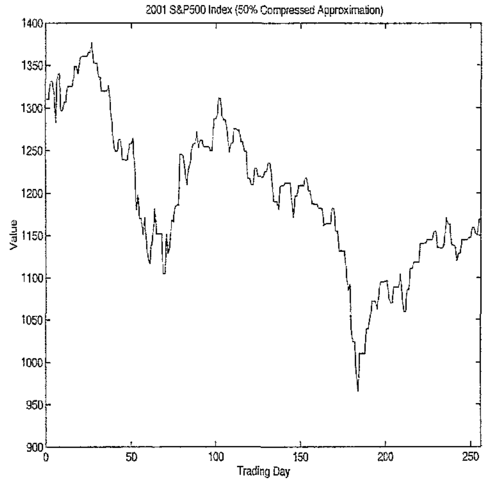

Figure 6: Compressed approximation using 50% of the coefficients.

of the compressed coefficients) and the resulting reconstruction has a global relative error of .739%. Finally, figure 9 is using about 10% of the original coefficients (about 20% of the compressed coefficients) and the resulting reconstruction has a global relative error of 1.355%. The results show the relationships to original compressed data size, additional size "culling" reconstruction queries, and relative error. In addition, Figure 10 is a different reconstruction, one that uses the 5 best (again measured with l^2 norms) levels to reconstruct the approximation. The resulting reconstruction has a global relative error of 1.623%. All of these simple univariate time series examples show that it is not difficult to achieve good compression with the approximation and still retain much of the important characteristics of the data, with respect to variation and change. This is in light of even further user query requested simplifications in the reconstruction. In order to test this procedure on actual data we have used simulation data of a can being crushed. We again show results using the familiar Haar orthonormal wavelet system. This data is time dependent and has about 13 independent variables per grid point. We show only a pressure field from the procedure due to space constraints although other other fields show similar behavior. Figures 11 and 14 show the original uncompressed can at the 1st and 30th time-steps of the simulation. Figures 13 and 16 show the compressed can at the same time-steps using 66% compression. Figures 12 and 15 show the compressed can at the same time-steps using 50% compression. The results show that while simulation data does not have the same simple reconstruction behavior of the univariate example above it is still possible to reconstruct data values and keep the kind of variational character in the results.

4. COMMENTS AND CONCLUSIONS

We have described research and development approaches we are taking to solve problems of knowledge discovery in large scale scientific simulation data. Our research adapts and extends ideas of wavelet theory to multivariate data, and formulates methodologies and algorithms for compressing the wavelet coefficients resulting from that work. We also devise ways in which users can effectively query the resulting compressed data in an intuitive fashion without understanding

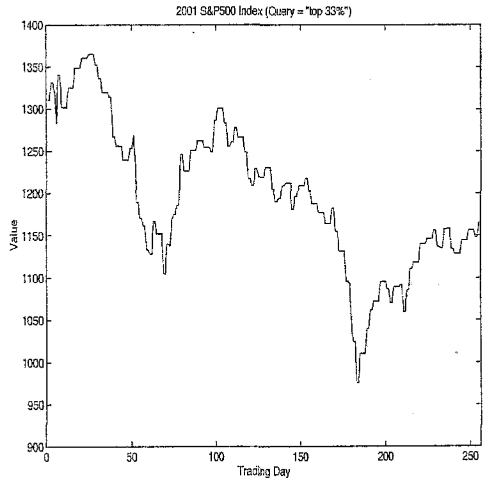

Figure 7: Reconstruction using 33% of the coefficients.

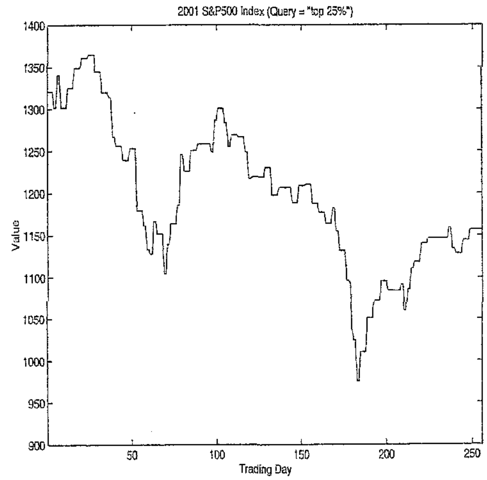

Figure 8: Reconstruction using 25% of the coefficients.

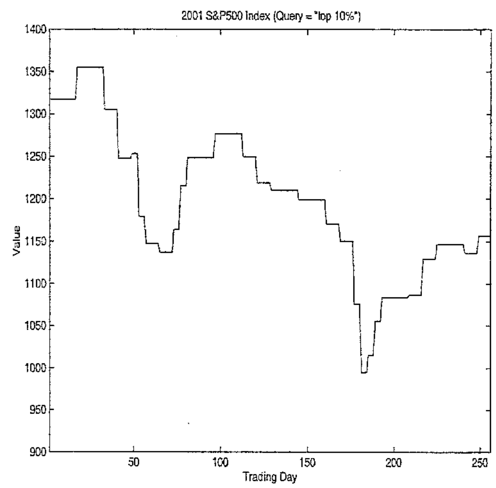

Figure 9: Reconstruction using 10% of the coefficients.

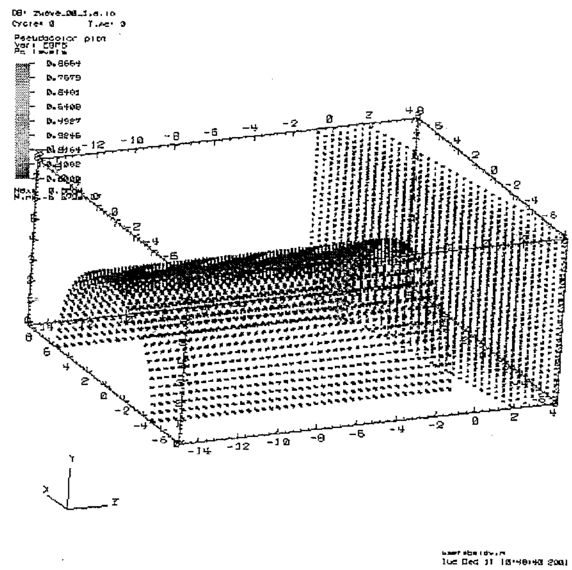

Figure 11: Time-step 1 from the original can dataset.

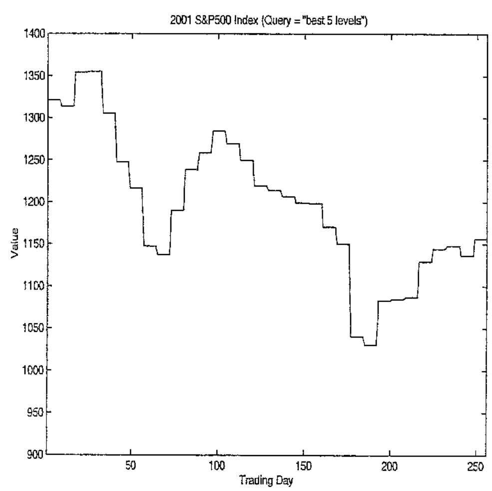

Figure 10: Reconstruction using the 5 most important levels.

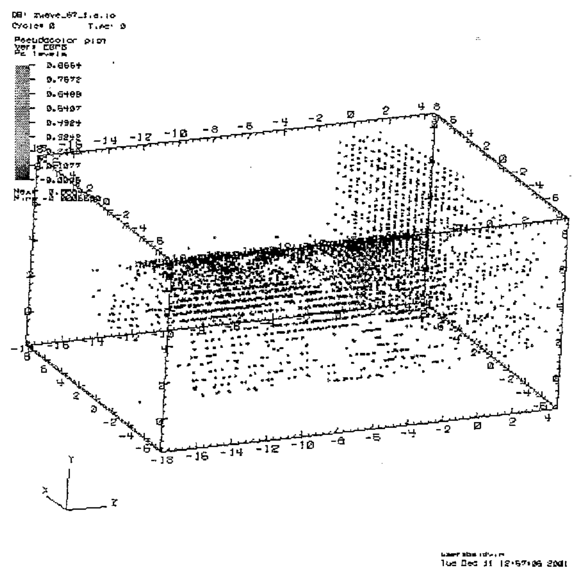

Figure 12: Time-step 1 from the 50% compressed can dataset.

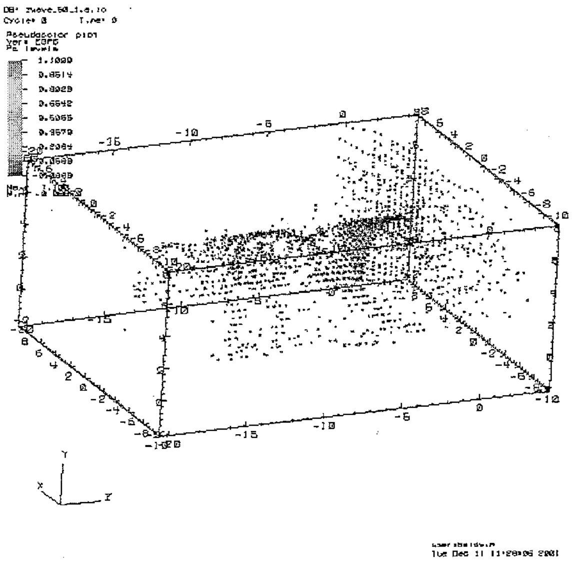

Figure 13: Time-step 1 from the 66% compressed can dataset.

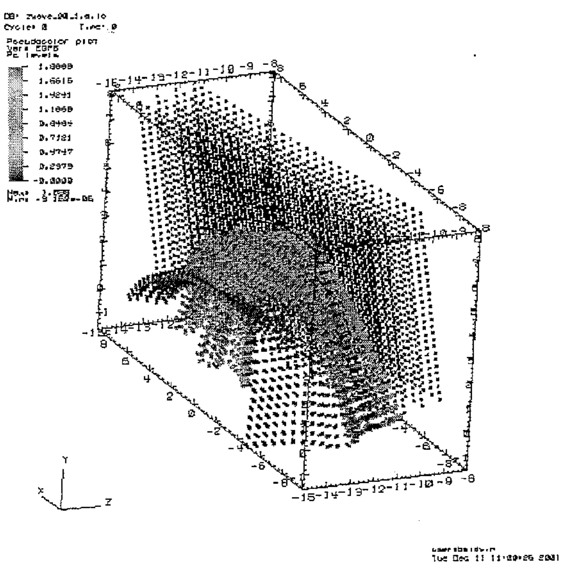

Figure 14: Time-step 30 from the original can dataset.

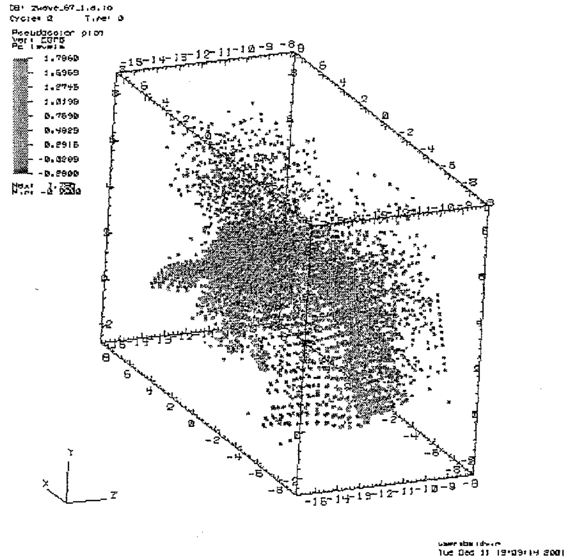

Figure 15: Time-step 30 from the 50% compressed can dataset.

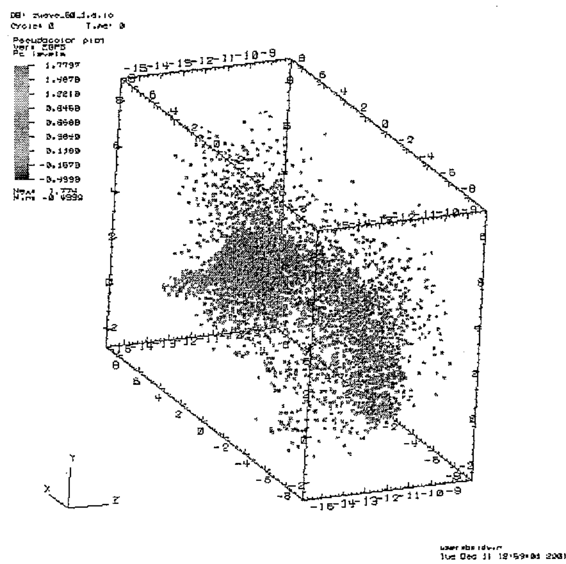

Figure 16: Time-step 30 from the 66% compressed can dataset.

too many of the wavelet specific details. In our initial experimentation we have found that using wavelets to decompose, compress, and reconstruct data yields results that are helpful in analyzing the dynamic portions of simulation data. The wavelets are attuned, in various degrees, to smoothness in data. Compressing by keeping only the largest coefficients implies that the reconstruction will be accurate around areas where the data is not smooth – highly dynamic. We intend to provide other models that address other inquiries about the data, such as range based queries, or perhaps more traditional clustering algorithms applied to the data. We view the wavelet queries as providing one of many tools that will allow a scientist or engineer to very quickly ascertain approximate characteristics of the data. As mentioned earlier this links compression with the data discovery process in a very significant way.

5. ACKNOWLEDGMENTS

This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

The authors would like to thank Kevin Durrenburger, Tina Eliassi-Rad, Roy Kamimura, Edward Smith, and Nu-Ai Tang from the Center for Applied Scientific Computing at the Lawrence Livermore National Laboratory for their advice and assistance reguarding this research.

6. REFERENCES

[1] G. Abdulla, C. Baldwin, T. Critchlow, R. Kamimura, I. Lozares, R. Music, N. A. Tang, B. S. Lee, and R. Snapp. Approximate ad-hoc query engine for simulation data. In *Joint Conference on Digital Libraries JCDL-01*, pages 255–256, June 2001.

[2] K. Chakrabarti, M. Garofalakis, R. Rastogi, and K. Shim. Approximate query answering using wavelets. *VLDB Journal*, 3, 2001.

[3] I. Daubechies. *Ten Lectures on Wavelets.* SIAM, 1992.

[4] R. DeVore, B. Jawerth, and B. Lucier. Image compression through wavelet transform coding. *IEEE Trans. Image Processing*, 38:719–746, 1992.

[5] R. DeVore, B. Jawerth, and V. Popov. Compression of wavelet decompositions. *American Journal of Mathematics*, 114:737–785, 1992.

[6] K. C. Eamonn Keogh, S. Mehrotra, and M. Pazzani. Locally adaptive dimensionality reduction for indexing large time series databases. In *2001 ACM SIGMOD Conference on Management of Data*, May 2001.

[7] U. M. Fayyad, D. Haussler, and P. E. Stolorz. KDD for science data analysis: Issues and examples. In *Knowledge Discovery and Data Mining*, pages 50–56, 1996.

[8] M. L. Hilton, B. D. Jawerth, and A. Sengupta. Compressing still and moving images with wavelets. *Multimedia Systems*, 2(5):218–227, 1994.

[9] B. Jawerth and W. Sweldens. An overview of wavelet based multiresolution analyses. *SIAM Rev.*, 36(3):377–412, 1994.

[10] Jpeg home page. http://www.jpeg.org/JPEG2000.htm, 1999.

[11] S. Mallat. *A Wavelet Tour of Signal Processing.* Academic Press, 1998.

[12] F. Murtagh, J.-L. Starck, and M. W. Berry. Overcoming the curse of dimensionality in clustering by means of the wavelet transform. *Computer Journal*, 43(2):107–120, 2000.

[13] R. T. Ogden. *Essential Wavelets for Statistical Applications and Data Analysis.* Springer Verlag, 1996.

[14] N. Ramakrishnan and A. Grama. Mining scientific data. In *Advances in Computers*, pages 119–169. Academic Press, 2001.

[15] R. R. Schmidt and C. Shahabi. Polap: A fast wavelet-based technique for progressive evaluation of olap queries. Technical Report 01-744, Department of Computer Science, University of Southern California, 2001.

[16] C. Shahabi, S. Chung, and M. Safar. A wavelet-based approach to improve the efficiency of multi-level surprise mining. In *PAKDD International Workshop on Mining Spatial and Temporal Data 2001*, 2001.

[17] G. Sheikholeslami, S. Chatterjee, and A. Zhang. WaveCluster: A multi-resolution clustering approach for very large spatial databases. In *Proceedings of the 24th Internation Conference Very Large Data Bases, VLDB*, pages 428–439, 24–27 1998.

[18] J.-L. Starck, F. Murtagh, and A. Bijaoui. *Image Processing and Data Analysis: The Multiscale Approach.* Cambridge University Press, 1998.

[19] G. Strang and T. Nguyen. *Wavelets and Filter Banks.* Wellesley Cambridge, 1996.

[20] D. S. Taubman and M. W. Marcellin. *JPEG2000 : Image Compression Fundamentals, Standards, and Practice.* Kluwer, 2001.

[21] A. Uhl. Wavelets and digital image compression I. Technical Report RIST++04/93, University of Salzburg, 1993.

[22] B. Vidakovic. *Statistical Modeling by Wavelets.* Wiley, 1999.

[23] D. Yu, S. Chatterjee, G. Sheikholeslami, and Z. Aidong. Efficiently detecting arbitrary shaped clusters in very large datasets with high dimensions. Technical Report 98-08, Department of Computer Science and Engineering, SUNY Buffalo, 1998.